

METHOD FOR PROCESSING CONTROLLER COMMUNICATION

Publication number: JP2001325115 (A)

Publication date: 2001-11-22

Inventor(s): SEWAI HIROSHI +

Applicant(s): SHIMADZU CORP +

Classification:

- international: G06F9/46; H04L12/28; G06F9/46; H04L12/28; (IPC1-7): G06F9/46; H04L12/28

- European:

Application number: JP20000145872 20000518

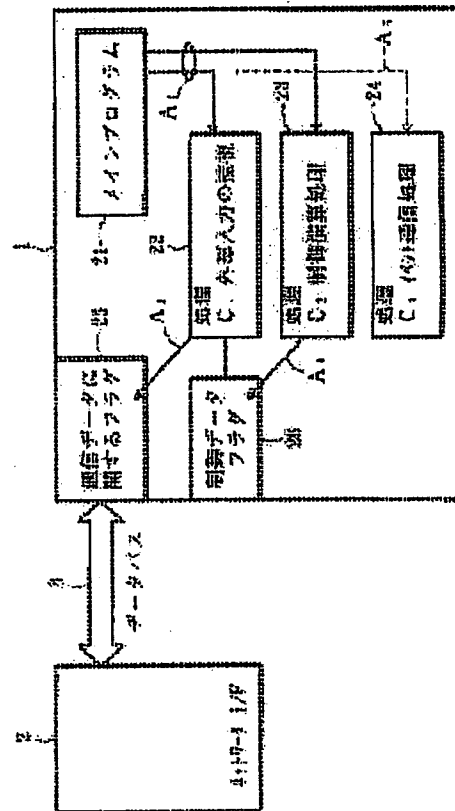
Priority number(s): JP20000145872 20000518

Also published as:

JP4283421 (B2)

Abstract of JP 2001325115 (A)

PROBLEM TO BE SOLVED: To improve responsiveness in the input of control data even in the case of a communication task started by an event such as the rewriting of a control program parameter having low priority, the monitoring of control operation information and the input of control data. **SOLUTION:** An external input monitoring part 22 checks a flag 25 concerned with communication data, checks a command when there are communication data and sets up a control data flag 26 when there are control data. A control operation processing part 23 checks the control data flag 26 before starting the operation processing of an application program, performs communication data processing concerned with the input of control data when the control data exist and returns the data to a network interface 2.



Data supplied from the *espacenet* database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-325115

(P2001-325115A)

(43) 公開日 平成13年11月22日 (2001.11.22)

(51) Int.Cl. ⁷	識別記号	F I	デマコード* (参考)
G 0 6 F 9/46	3 4 0	C 0 6 F 9/46	3 4 0 B 5 B 0 9 8
H 0 4 L 12/28		H 0 4 L 11/00	3 1 0 Z 5 K 0 3 3

審査請求 未請求 請求項の数 1 O L (全 5 頁)

(21) 出願番号 特願2000-145872(P2000-145872)

(22) 出願日 平成12年5月18日 (2000.5.18)

(71) 出願人 000001993

株式会社島津製作所

京都府京都市中京区西ノ京桑原町1番地

(72) 発明者 瀬和居 裕志

京都市中京区西ノ京桑原町1番地 株式会

社島津製作所内

(74) 代理人 100084962

弁理士 中村 茂信

Fターム(参考) 5B098 AA09 CC01 CC07 FF08 GA04

5K033 AA02 BA03 BA08 DA01 DB12

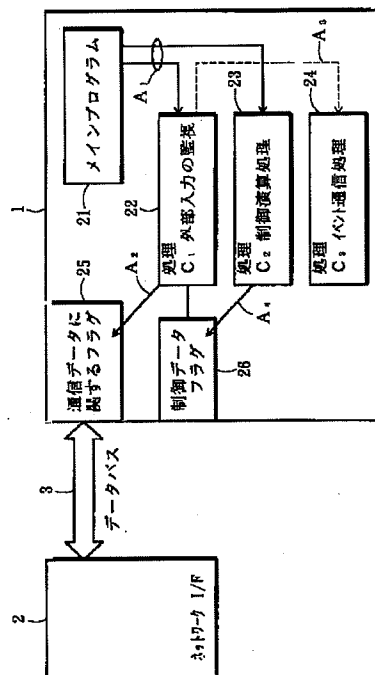
DB16

(54) 【発明の名称】 コントローラ通信処理方法

(57) 【要約】

【課題】 優先性の低い制御プログラム・パラメータの書き換え、制御演算情報のモニタリング、制御データの取り込み等のイベント起動される通信タスクでも、制御データの取り込みに関しては応答性を向上する。

【解決手段】 外部入力監視部22で通信データに関するフラグ25をチェックし、通信データ有りの場合にコマンドをチェックし、制御データ有りの場合に制御データフラグ26を立てる。制御演算処理部23ではアプリケーションプログラムの演算処理に入る前に、制御データフラグ26をチェックし、制御データ有りの場合に制御データの取り込みに関する通信データ処理を行い、ネットワークインタフェース2にデータを返送する。



【特許請求の範囲】

【請求項1】外部入力 of 監視と、制御演算処理を行う定周期タスクと、制御プログラム・パラメータの書き換え、制御演算情報のモニタリング、制御データの取り込み等のためのイベント起動タスクとを設けるコントローラの通信処理方法において、前記外部入力 of 監視で通信データの有無の監視をしており、通信データのコマンドの内容をチェックして処理を行うべきタスクの振り分けを行うとともに、前記定周期タスクにも前記イベントタスク機能を持たせるようにしたことを特徴とするコントローラの通信処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、プロセス制御装置等のコントローラの通信処理方法に関する。

【0002】

【従来の技術】一般に、プロセス制御装置は、図4に示すように、コントローラ1-1、1-2、……がデータバス3を介してネットワークインタフェース2に接続され、このネットワークインタフェース2は制御LAN4でMMI（マンマシンインタフェース）5に接続されている。

【0003】この種の制御装置における通信データ処理の流れは、図5に示す通りであり、先ずMMI5より制御LAN4を通じて、a. コントローラ1にデータ（制御データ）を送信する。ネットワークインタフェース2では、b. コントローラ1にデータを振り分け、コントローラ1上の「通信データ有り」のフラグをONする。コントローラ1では、下記のCの処理を行う。先ず、第1にC₁の処理として、図6に示すように、通信データ有りを判定して（ステップST1）、通信データ有りの場合にC₃の処理を起動する（ステップST2）。この起動後、あるいはステップST1で通信データ無しの場合に、その他の処理（ステップST3）を経て、C₁の処理を終了する。

【0004】C₂の処理は、図7で示すように、定周期でユーザアプリケーションとして設定された演算ブロック順に演算処理を行う（ステップST4）。C₁処理のステップST2にてC₃の処理が起動されると、図8に示すように通信データを読み出し（ステップST5）、コマンドに従い、必要な処理（C₂に制御データを流す）を行う（ステップST6）。C₃でのこのような処理を行うのは、C₂処理で行う制御演算処理の中には、図11で示すように、MMI5からの制御データに基づき、演算処理を行うブロック6が用意されており、ユーザアプリケーションで設定することができる。このようなブロックに対しても、MMI5からの取り込みに関しては、C₃で行うことにしていることによる。そして、ネットワークインタフェース2にデータを返送する（ステップST7）。

【0005】ネットワークインタフェース2では、d. MMI5にコントローラ1からの通信を転送する。MMI5では、e. コントローラ1からの通信を確認する。

【0006】以上のように、従来のコントローラではC₁：外部入力 of 監視と、C₂：制御演算処理を行う定周期タスクと、C₃：制御プログラム・パラメータの書き換え、制御演算情報のモニタリング、制御データの取り込み等のためのイベント起動される通信タスクとを設けている。コントローラの性質上、制御演算の定周期性を保つことがもっとも重要なため、C₁、C₂の各タスクの優先順位をC₃より上げて処理を行っている。

【0007】

【発明が解決しようとする課題】上記した従来のコントローラの処理において、C₂の定周期タスクはユーザが設定したアプリケーションプログラムを実行する部分であり、コントローラの処理負荷はC₂の処理の大きさ、つまりアプリケーションプログラムのサイズで決まる。各タスクの優先順位の関係から、C₂の処理が大きくなるとC₃のイベント起動されるタスクに対する応答が悪くなる場合が起こりうる。C₃で行われるべき処理の中に制御演算上必要となる情報が含まれる場合、応答不良に起因する誤操作につながることも考えられる、という問題がある。

【0008】この発明は上記問題点に着目してなされたものであって、従来のコントローラのC₃で行われる処理の中でも制御データの取り込みに関するイベント通信処理に対応する応答性向上を図ることを課題としている。

【0009】

【課題を解決するための手段】この発明のコントローラの通信処理方法は、外部入力 of 監視と、制御演算処理を行う定周期タスクと、制御プログラム・パラメータの書き換え、制御演算情報のモニタリング、制御データの取り込み等のためのイベント起動タスクとを設けるコントローラの通信処理方法において、前記外部入力 of 監視で通信データの有無の監視をしており、通信データのコマンドの内容をチェックして処理を行うべきタスクの振り分けを行うとともに、前記定周期タスクにも前記イベントタスク機能を持たせるようにしている。

【0010】この通信処理方法では、従来のコントローラでは一律C₃で行っていた通信処理のうち、制御データの取り込みに関する処理は、C₂の定周期処理と関連付けられている。そのため、C₂の中にも通信に関する処理が加えられ、定周期で制御データの受信に関する処理が行われる。C₂での通信処理が制御演算処理の負荷に影響を与えることを極力避けるために、制御データの取り込みに関するコマンドのみに対応可能としている。

【0011】

【発明の実施の形態】以下、実施の形態により、この発明をさらに詳細に説明する。図9は従来のコントローラ

1の機能構成を示すブロック図である。このコントローラ1は、メインプログラム11と、外部入力監視部12と、制御演算処理部13と、イベント通信処理部14と、通信データに関するフラグ15とを備えている。外部入力監視部12は、メインプログラム11からの定周期の起動 A_1 で図6に示す C_1 の処理を実行し、監視 A_2 を行う。また、制御演算処理部13はメインプログラム11からの定周期の起動 A_1 で図4に示す C_2 の処理を、イベント通信部14は外部入力監視部12からの起動 A_3 で図8に示す C_3 の処理を実行する。

【0012】このコントローラ1の一連の処理動作例を図10のタイムチャートを参照して説明する。図10において、 C_1 は周期 T_1 で外部入力等、コントローラH/Wに関する情報の監視、処理を行う。 C_2 は周期 T_2 ($=n \times T_1$)で定期演算処理を行う。 C_2 は C_1 よりも優先度を低くしてあり、処理が完結するまでの間、 C_1 が割り込む場合もある。 C_3 は C_1 において「通信データON」の場合に起動される。優先度は C_1 、 C_2 よりも低くしてあり、 C_1 、 C_2 の空き時間に処理する。

【0013】 C_2 の処理は、ユーザアプリケーションにより増加する余地があるため、 C_2 の処理が大きくなると C_3 の処理に対する余裕が小さくなり、コントローラ1の外から見て、イベント通信に対する応答が悪くなる。

【0014】図1は、この発明の一実施形態コントローラ1の機能構成を示すブロック図である。この実施形態コントローラ1はメインプログラム21と、外部入力監視部22と、制御演算処理部23と、イベント通信処理部24と、通信データに関するフラグ25と、制御データフラグ26を備えており、上記した従来のコントローラの不具合を解消する構成となっている。

【0015】外部入力監視部22は、メインプログラム21からの定周期の起動 A_1 で図2に示す C_1 の処理を実行し、また監視 A_2 を行う。制御演算処理部23は、メインプログラム21からの定周期の起動 A_1 で図3に示す C_2 の処理を実行する。また C_2 の処理でチェック A_4 を行う。イベント通信処理部24は、外部入力監視部22よりの起動 A_3 で図8に示した従来の C_3 とほぼ同様の処理を実行し、また監視 A_2 を行う。この発明の実施形態コントローラは、外部入力監視部22と制御演算処理部23で実行する C_1 の処理と C_2 の処理に特徴がある。

【0016】この実施形態コントローラ1では、定周期処理で図10の T_1 毎に、 C_1 の処理を開始し、先ず通信データフラグ25をチェックし、通信データ有りが判定する(ステップST11)。通信データがなければ、その他の処理を経て(ステップST16)、 C_1 の処理を終了する。ステップST11において、通信データ有りの場合は、コマンドの内容をチェックし(ステップST12)、制御データ有りが否か判定する(ステップST13)。制御データでない場合は、従来と同様、 C_3 の

処理を起動し(ステップST15)、その他の処理(ステップST16)を経て、 C_1 の処理を終了する。

【0017】ステップST13で、制御データ有りの場合は、 C_2 の定周期処理で制御データに関する処理を行わせるため、「制御データ有り」のフラグ26を C_2 に付す(ステップST14)。その後、その他の処理を経て(ステップST16)、 C_1 の処理を終了する。

【0018】この実施形態コントローラ1でも図10に示す $n_2 = n_1 \times T_1$ 毎の定周期で、 C_2 の処理を実行する。 C_2 の処理開始で、先ず「制御データ有り」のフラグ26を参照して、制御データ有りの判定を行う(ステップST21)。図2のステップST14で説明したように、 C_1 の処理で「制御データ有り」のフラグ26が立っていると、本来の制御演算処理の前に制御の取り込みに関する通信データ処理を行う。すなわち、制御データの取り込みに関する通信データ処理を行う(ステップST22)とともに、ネットワークインタフェース2にデータを返送し(ステップST23)、その後、ユーザアプリケーションとして設定された演算ブロックの順に演算処理、つまり本来の制御演算処理を行う(ステップST24)。ステップST21で「制御データ有り」のフラグ26が立っていない場合は、本来の制御演算処理を実行して(ステップST24)、 C_2 の処理を終了する。

【0019】

【発明の効果】この発明によれば、外部入力監視で通信データの有無を監視しており、通信データのコマンドの内容をチェックして、処理を行うべきタスクの振り分けを行うとともに、制御演算処理の定周期タスクにもイベントタスクの機能を持たせるようにしているので、コントローラの負荷、つまりアプリケーションプログラムに関する処理量が増大しても、制御データの取り込みに関する応答性を確保することができる。

【図面の簡単な説明】

【図1】この発明の実施形態コントローラの機能構成を示すブロック図である。

【図2】同実施形態コントローラにおける外部入力監視処理を説明するフロー図である。

【図3】同実施形態コントローラの定周期制御演算処理を説明するフロー図である。

【図4】一般的なプロセス制御装置の構成を示すブロック図である。

【図5】同プロセス制御装置におけるデータ処理の流れを説明する図である。

【図6】従来のコントローラの外部入力監視の処理を説明するフロー図である。

【図7】従来のコントローラの定周期の制御演算処理を説明するフロー図である。

【図8】従来のコントローラのイベント通信処理を説明するフロー図である。

【図9】従来のコントローラの機能構成を示すブロック図である。

【図10】従来のコントローラのタスクの優先順位を説明するためのタイムチャートである。

【図11】従来のコントローラにおける制御データ処理について説明する図である。

【符号の説明】

1 コントローラ

2 ネットワークインタフェース

3 データバス

21 メインプログラム

22 外部入力監視部

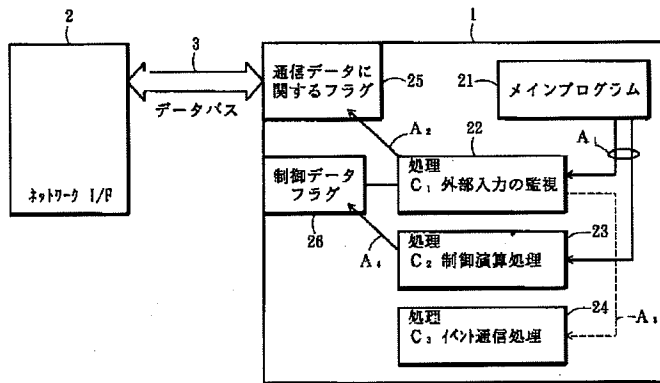
23 制御演算処理部

24 イベント通信処理部

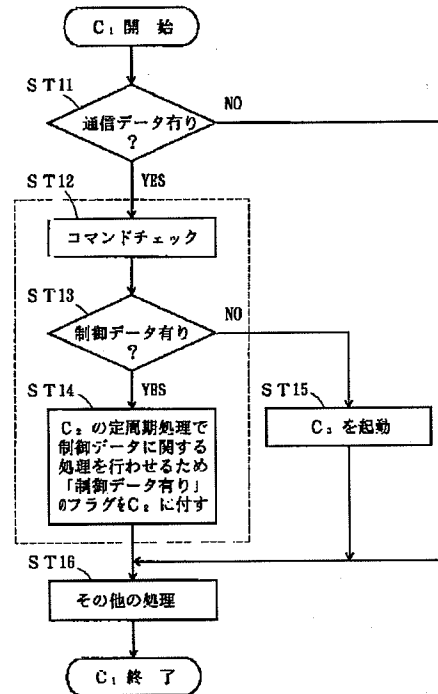
25 通信データに関するフラグ

26 制御データフラグ

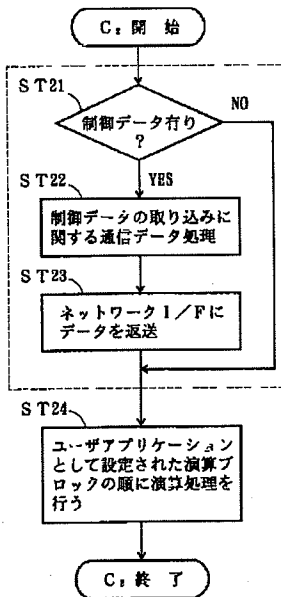
【図1】



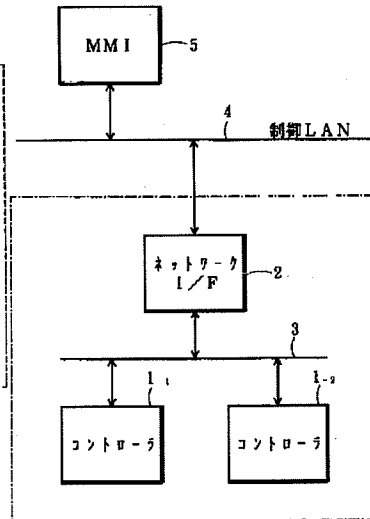
【図2】



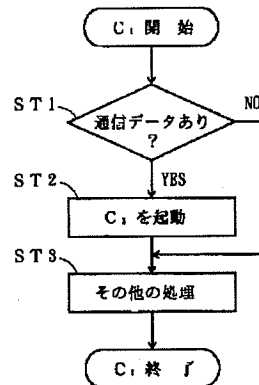
【図3】



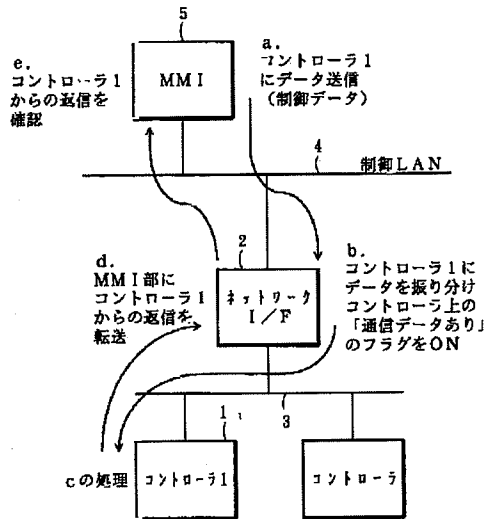
【図4】



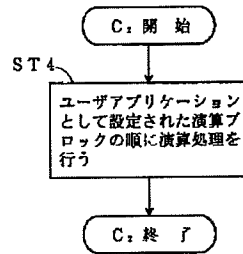
【図6】



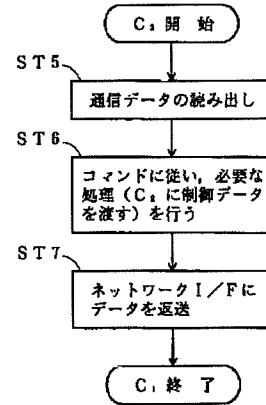
【図5】



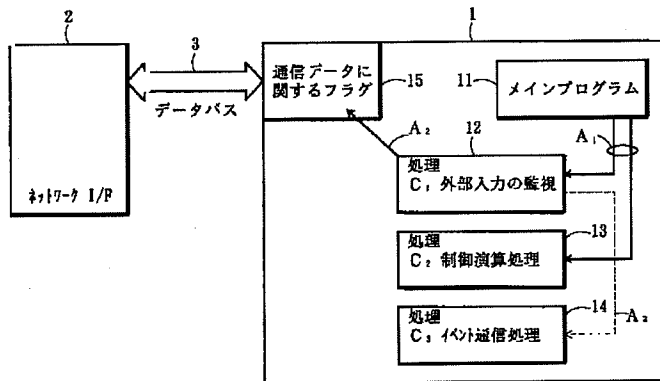
【図7】



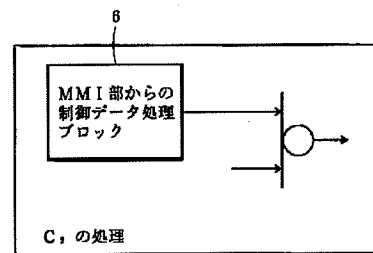
【図8】



【図9】



【図11】



【図10】

